

**IN THE CLAIMS**

Please amend the claims to read as indicated herein.

1. (currently amended) A system comprising:

a module that ~~embeds a value for an attribute into a key to data~~ provides a key that includes (1) a unique identifier of a datum, and (2) an additional attribute; and  
a module that ~~creates an index of said data and associates said data with said key,~~  
wherein said key is utilized by a process that associates said key with said datum in an index of said datum, so that said additional attribute is also associated with said datum in said index.

2. (currently amended) The system of claim 1, wherein said ~~key comprises an identifier, and said value~~ additional attribute is encoded into said unique identifier.

3. (currently amended) The system of claim 1, wherein said ~~data comprises~~ datum is one of a plurality of digital documents.

4. (currently amended) The system of claim 1, wherein said ~~data~~ datum is stored in a database.

5 – 8. (canceled)

9. (currently amended) A method, executed by a processing system, comprising:  
~~embedding a value for an attribute into a key to data~~ creating a key that includes (1)  
a unique identifier of a datum, and (2) an additional attribute; and  
~~creating an index of said data; and in which said key is associated with said datum,~~  
so that said additional attribute is also associated with said datum in said  
index  
~~associating said data with said key.~~

10. (currently amended) The method of claim 9, wherein said ~~data comprises an identifier, and said value~~ additional attribute is encoded into said unique identifier.

11. (currently amended) The method of claim 9, wherein said ~~data comprises~~ datum is one of a plurality of digital documents.

12. (currently amended) The method of claim 9, further comprising storing said ~~data~~ datum in a database.

13. (currently amended) The method of claim 9, wherein said datum is one of a plurality of data ~~is that are~~ distributed over a plurality of network nodes.

14. (currently amended) The method of claim 13, wherein said ~~embedding creating~~ is performed ~~over said plurality of network nodes~~ for said plurality of data.

15. (canceled)

16. (currently amended) A software program, stored on a data carrier, for executing a method when run on a data processing system, said method comprising:

~~embedding a value for an attribute into a key to data~~ creating a key that includes (1)

a unique identifier of a datum, and (2) an additional attribute; and

~~creating an index of said data; and~~ in which said key is associated with said datum,

so that said additional attribute is also associated with said datum in said

index

~~associating said data with said key.~~

17. (currently amended) The software program of claim 16, wherein said ~~data~~ comprises an identifier, and said value additional attribute is encoded into said unique identifier.

18. (currently amended) The software program of claim 16, wherein said datum is one of a plurality of data-is that are distributed over a plurality of network nodes.

19. (currently amended) The software program of claim 18, wherein said embedding creating is performed ~~over said plurality of network nodes~~ for said plurality of data.

20. (canceled)

21. (currently amended) The system of claim 1,  
wherein said key is a first key and said ~~value is~~ additional attribute has a first value,  
and  
wherein said system further comprises:

a module that searches said index and obtains a list that includes said first key  
and a second key, wherein said second key has a second value  
~~embedded therein~~ for said additional attribute; and  
a module that sorts said list in an order determined by said first value and said  
second value.

22. (currently amended) The system of claim 21, wherein said module that ~~embeds~~  
~~said value~~ provides said key and said module that searches said index are connected by a  
network having a plurality of nodes.

23. (currently amended) The system of claim 22, wherein said datum is one of a  
plurality of data-is that are distributed over said plurality of nodes.

24. (currently amended) The system of claim 22, wherein said module that ~~embeds~~  
~~said value~~ provides said key is distributed over said plurality of nodes.

25. (currently amended) The system of claim 22, wherein ~~the~~ said module that  
searches said index is distributed over said plurality of nodes.

26. (currently amended) The method of claim 9,  
wherein said key is a first key and said ~~value is~~ additional attribute has a first value,  
and  
wherein said method further comprises:  
searching said index;  
obtaining a list that includes said first key and a second key, wherein said  
second key has a second value ~~embedded therein~~ for said additional  
attribute; and  
sorting said list in an order determined by said first value and said second  
value.

27. (currently amended) The method of claim 26,  
wherein said datum is one of a plurality of data-is that are distributed over a  
plurality of network nodes; and  
wherein said searching is performed over said plurality of network nodes.

28. (currently amended) The software program of claim 16,  
wherein said key is a first key and said ~~value is~~ additional attribute has a first value,  
and  
wherein said method further comprises:  
searching said index;  
obtaining a list that includes said first key and a second key, wherein said  
second key has a second value ~~embedded therein~~ for said additional  
attribute; and  
sorting said list in an order determined by said first value and said second  
value.

29. (currently amended) The software program of claim 28,  
wherein said datum is one of a plurality of data-is that are distributed over a  
plurality of network nodes, and

wherein said searching is performed over said plurality of network nodes.

30. (currently amended) A system comprising:

a module that:

- (A) searches an index of data, wherein said data includes:
    - (i) a ~~first item of data~~ datum associated with a first key ~~having embedded therein that includes:~~
      - (1) a unique identifier of said first datum, and
      - (2) an additional attribute having a first value for an attribute,and
    - (ii) a ~~second item of data~~ datum associated with a second key ~~having embedded therein that includes:~~
      - (1) a unique identifier of said second datum, and
      - (2) an additional attribute having a second value for said attribute, and
  - (B) obtains a list that includes said first key and said second key; and
- a module that sorts said list in an order determined by said first value and said second value.

31. (currently amended) A method, executed by a processing system, comprising:  
searching an index of data, wherein said data includes:

- (i) a ~~first item of data~~ datum associated with a first key ~~having embedded therein that includes:~~
  - (1) a unique identifier of said first datum, and
  - (2) an additional attribute having a first value for an attribute, and
- (ii) a ~~second item of data~~ datum associated with a second key ~~having embedded therein that includes:~~
  - (1) a unique identifier of said second datum, and
  - (2) an additional attribute having a second value for said attribute;

obtaining a list that includes said first key and said second key; and  
sorting said list in an order determined by said first value and said second value.

32. (currently amended) A software program, stored on a data carrier, for executing a method when run on a data processing system, said method comprising:

searching an index of data, wherein said data includes:

(i) a ~~first item of data~~ datum associated with a first key ~~having embedded therein that includes:~~

(1) a unique identifier of said first datum, and

(2) an additional attribute having a first value for an attribute, and

(ii) a ~~second item of data~~ datum associated with a second key ~~having embedded therein that includes:~~

(1) a unique identifier of said second datum, and

(2) an additional attribute having a second value for said attribute;

obtaining a list that includes said first key and said second key; and

sorting said list in an order determined by said first value and said second value.